



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,403	01/04/2002	Benjamin T. Sander	5500-74800	7348
7590	09/21/2004			
			EXAMINER	
			HUISMAN, DAVID J	
			ART UNIT	PAPER NUMBER
			2183	

DATE MAILED: 09/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	10/037,403	SANDER, BENJAMIN T. <i>fl 8</i>
	Examiner David J. Huisman	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)  Responsive to communication(s) filed on 04 January 2002.

2a)  This action is **FINAL**.                            2b)  This action is non-final.

3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## **Disposition of Claims**

4)  Claim(s) 1-35 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5)  Claim(s) \_\_\_\_\_ is/are allowed.  
6)  Claim(s) 1-35 is/are rejected.  
7)  Claim(s) \_\_\_\_\_ is/are objected to.  
8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on 04 January 2004 is/are: a)  accepted or b)  objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 15 August 2002.

4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.

5)  Notice of Informal Patent Application (PTO-152)

6)  Other: \_\_\_\_\_.

## **DETAILED ACTION**

1. Claims 1-35 have been examined.

### ***Papers Submitted***

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 8/15/2002.

### ***Specification***

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.
4. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

### ***Drawings***

5. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the pointer of claim 13, the plurality of pointers of claim 14, and the stack tags of claim 15 must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure

number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency.

Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

### ***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-3, 5-6, 9-10, 18-23, 27-28, and 32-33 are rejected under 35

U.S.C. 102(b) as being anticipated by Gochee, U.S. Patent No. 5,732,272.

8. Referring to claim 1, Gochee has taught a return address mechanism comprising:

a) a return storage, wherein the return storage comprises a first entry, wherein the first entry comprises a count and a first return address that corresponds to a recently detected call operation. See Fig. 6, and note that the top stack entry (block #3) includes a counter.

Also, see column 7, lines 30-33, and note that each entry is associated with a different subroutine return address.

b) a controller coupled to the return storage and configured to receive a new return address corresponding to a most recently detected call instruction, wherein the controller is further configured to compare the new return address to the first return address, wherein if the new return address equals the first return address, the controller is configured to increase a value of the count. See Fig.5 and column 7, lines 8-16. Note that when a call occurs, it is checked if the return address is already on the stack. If so, a counter is incremented.

It should be noted that the recitation "prediction mechanism" has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

9. Referring to claim 2, Gochee has taught a mechanism as described in claim 1. Gochee has further taught that if the new return address does not equal the first return address, the controller is configured to allocate a new entry for the new return address. See Fig.5 and column 6, line 64, to column 7, line 7. Note that if the new return address is not at the top of the stack, a new entry is allocated for it. And, as seen in column 7, lines 30-33, each entry in the stack belongs to a different subroutine return address. Therefore, if the new address is not already in the stack, then a new entry is allocated.

Art Unit: 2183

10. Referring to claim 3, Gochee has taught a mechanism as described in claim 2. Gochee has further taught that the controller is configured to initialize a count in the new entry to a minimum count value. See column 8, lines 13-18. Note that the counter is initially one, indicating that the associated subroutine has been invoked once.
11. Referring to claim 5, Gochee has taught a mechanism as described in claim 1. Gochee has further taught that the return storage is implemented as a stack structure, and wherein the first entry is identified by a top of stack pointer. See Fig.6 and note the top-of-stack pointer SP which points to the first entry.
12. Referring to claim 6, Gochee has taught a mechanism as described in claim 5. Gochee has further taught that if the new return address does not equal the first return address, the controller is configured to allocate a new entry for the new return address and to modify the top of stack pointer to identify the new entry. See Fig.5, step 508 and note that when a new return address is encountered for a first time, an entry is allocated at the top of the stack. Furthermore, it is the inherent nature of a stack that when a new entry is allocated (added to the top of the stack), the top of stack pointer must also be updated so that it points to the new top of the stack.
13. Referring to claim 9, Gochee has taught a mechanism as described in claim 1. Gochee has further taught that each entry in the return storage comprises a respective count. See Fig.6.
14. Referring to claim 10, Gochee has taught a mechanism as described in claim 1. Gochee has further taught that fewer than all entries in the return storage comprise a respective count. More specifically, not all return addresses will be encountered twice.

Therefore, for those returns, there will be no respective count associated with it where the count represents the number of times the same return address is encountered (repeated).

15. Referring to claim 18, Gochee has taught a method comprising:
  - a) detecting a call instruction in an instruction stream and in response to said detecting, comparing a first return address stored in a first entry in a return storage to a second return address that is associated with the call instruction. When a call instruction is encountered, the process in Fig.5 is followed, and this includes either allocating a new stack entry if it is a new return address.
  - b) if the first return address equals the second return address, increasing a first count associated with the first entry. See Fig.5 and column 7, lines 8-16. Note that when a call occurs, it is checked if the return address is already on the stack. If so, a counter is incremented.

It should be noted that the recitation “method of predicting return addresses” has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

16. Referring to claim 19, Gochee has taught a method as described in claim 18. Gochee has further taught allocating a second entry to the second return address in the return storage if the second return address does not equal the first return address. See

Fig.5 and note that if it's a new return address (second address not equal to the first address), then a new entry is allocated.

17. Referring to claim 20, Gochee has taught a method as described in claim 19. Gochee has further taught that the controller is configured to initialize a second count associated with the second entry to a minimum count value. See column 8, lines 13-18. Note that the counter is initially one, indicating that the associated subroutine has been invoked once.

18. Referring to claim 21, Gochee has taught a method as described in claim 19. Gochee has further taught providing the second return address and decreasing a second count associated with the second entry in response to a return instruction being detected. When a return instruction is detected, the process in Fig.7 is followed, and it can be seen that in steps 706-708, a return address is provided and a counter is decremented.

19. Referring to claim 22, Gochee has taught a method as described in claim 21. Gochee has further taught removing the second entry from the return storage if the value of the second count associated with the second return address is equal to a minimum value as a result of said decreasing. See Fig.7, steps 714 and 716. That is, if the counter is down to the original (minimum) value, then the last return is being processed, thereby allowing the entry in the stack to be removed.

20. Referring to claim 23, Gochee has taught a method as described in claim 18. Gochee has further taught that said increasing comprises incrementing a counter. See step 510 in Fig.5.

21. Referring to claim 27, Gochee has taught a method as described in claim 18. Gochee has further taught that the return storage is implemented as a stack structure, and

wherein a top entry in the return storage is identified by a top of stack pointer. See Fig.6 and note the top-of-stack pointer SP which points to the top entry.

22. Referring to claim 28, Gochee has taught a method as described in claim 27. Gochee has further taught modifying the top of stack pointer to identify a second entry in the return storage allocated to the second return address if the second return address does not equal the first return address. See Fig.5, step 508 and note that when a new return address is encountered for a first time, an entry is allocated at the top of the stack. Furthermore, it is the inherent nature of a stack that when a new entry is allocated (added to the top of the stack), the top of stack pointer must also be updated so that it points to the new top of the stack.

23. Referring to claim 32, Gochee has taught a method as described in claim 18. Gochee has further taught that each entry in the return storage has an associated count. See Fig.6.

24. Referring to claim 33, Gochee has taught a method as described in claim 18. Gochee has further taught that fewer than all entries in the return storage have an associated count. More specifically, not all return addresses will be encountered twice. Therefore, for those returns, there will be no respective count associated with it where the count represents the number of times the same return address is encountered (repeated).

25. Claim 35 is rejected under 35 U.S.C. 102(e) as being anticipated by Poplingher, U.S. Patent No. 6,170,054.

26. Referring to claim 35, Poplingher has taught a method of predicting a return address, the method comprising:

a) detecting a return instruction in an instruction stream and in response to said detecting, providing a first return address stored in a return storage as a predicted return address and decreasing a count associated with the first return address. See Fig. 5, steps 510 and 512, note that if a return instruction is detected and predicted, then an age associated with that return instruction (Fig.3 and Fig.4) will be decremented.

***Claim Rejections - 35 USC § 103***

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

28. Claims 4, 14, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gochee, as applied above.

29. Referring to claim 4, Gochee has taught a mechanism as described in claim 1. Gochee has not explicitly taught that if the new return address equals the first return address and the value of the count equals a maximum count value, the controller is configured to allocate a new entry for the new return address. However, a person of ordinary skill in the art would have recognized that the counter of Gochee would have a maximum because the counter is finite (determined by the hardware design). And, incrementing a finite counter at its maximum value will result in overflow, thereby corrupting the counter value (for instance, if the maximum value of the counter is seven (111 in binary) and the system encounters the same return address for the eighth time, then incrementing the counter will result in 000, thereby indicating that zero returns have

been encountered as opposed to eight). It should be realized that a particular call instruction may be executed a number of times greater than the counter can accommodate due to the fact that the execution of the call instruction is largely based on variables determined during runtime. Clearly, it is desirable to prevent corruption and accommodate the repeated execution of the same call instruction, and therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee such that a new entry is allocated if a counter is at its maximum value and it requires further incrementing.

30. Referring to claim 14, Gochee has taught a mechanism as described in claim 1. Gochee has not explicitly taught that the first entry comprises a plurality of pointers to next entries in the return storage, wherein each of the pointers is associated with a particular value of the count. However, Official Notice is taken that it is well known and accepted than a stack may be implemented via doubly-linked list (DLL). With a DLL, each entry (node) has a pointer to the next entry and to the previous entry, whereas regular linked lists have only a single pointer. Consequently, with a DLL, access is faster to the previous node which is not pointed to in a single linked list. And, as can be seen in Fig.6 of Gochee, each entry (node) is associated with a different counter value. Therefore, in order to achieve the benefits of a DLL, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee such that each entry has multiple pointers to next entries.

31. Referring to claim 26, Gochee has taught a method as described in claim 18. Further, claim 26 is rejected for the same reasons set forth in the rejection of claim 4.

Art Unit: 2183

32. Claims 7-8, 11-12, 16-17, 24-25, 29-31, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gochee, as applied above, in view of Hoyt et al., U.S. Patent No. 5,768,576 (herein referred to as Hoyt).

33. Referring to claim 7, Gochee has taught a mechanism as described in claim 5. Gochee has not taught that in response to a branch prediction being made, the controller is configured to save a copy of a current value of the top of stack pointer. However, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction. In addition, Hoyt has taught that if a branch is mispredicted, the system needs to restart execution from the appropriate point (the system will need to be restored to a previous state). See column 2, lines 5-11. Furthermore, claims 6 and 7 of Hoyt show that in order to perform this restoration, the top of stack pointer must be saved. Consequently, in order to perform this restoration, which is required if branch prediction is employed, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to save the pointer when a prediction is made.

34. Referring to claim 8, Gochee in view of Hoyt has taught a mechanism as described in claim 7. Furthermore, recall that Hoyt has taught that when a prediction is made, if a misprediction results, the system must be restored to its previous state. Consequently, in order to restore the system's previous state, the previous state must be saved. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee such that the controller is further configured to

save a copy of the value of the count associated with the first entry if the first entry is identified by the top of stack pointer when the branch prediction is made.

35. Referring to claim 11, Gochee has taught a mechanism as described in claim 1. Gochee has further taught that in response to a return instruction being detected, the controller is configured to provide the most recently detected return address and to decrease a value of a count associated with the most recently detected return address. See Fig. 7, steps 706, 708, and 710. Gochee has not taught that this return address a predicted return address. However, return address prediction is well known and expected in the art, and further supported by Hoyt. More specifically, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction, wherein the return address is a predicted return address.

36. Referring to claim 12, Gochee has taught a mechanism as described in claim 11. Gochee has further taught that if the value of the count associated with the most recently detected return address is equal to a minimum value after being decreased, the controller is further configured to remove the most recently detected return address's entry from the return storage. See Fig. 7, steps 714 and 716. That is, if the counter is down to the original (minimum) value, then the last return is being processed, thereby allowing the entry in the stack to be removed.

37. Referring to claim 16, Gochee has taught a return address mechanism comprising:

a) a return storage comprising a first entry wherein the first entry comprises a count and a first return address that corresponds to a most recently detected call operation. See Fig. 6, and note that the top stack entry (block #3) includes a counter. Also, see column 7, lines 30-33, and note that each entry is associated with a different subroutine return address.

b) a controller coupled to the return storage and configured to provide a return address, wherein if the count indicates that the first return address corresponds to more than one call operation, the controller is configured to provide the return address by providing the first return address and decreasing the count. See Fig. 7, steps 706-710.

c) Gochee has not explicitly taught that the return address is a predicted return address. However, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction.

38. Referring to claim 17, Gochee has taught a mechanism as described in claim 16. Gochee has further taught that if the count indicates that the first return address corresponds to a single call operation, the controller is configured to provide the return address by providing the first return address and removing the first entry from the return storage. See Fig. 7, steps 714 and 716. Note that if the counter is down to its original (minimum) value, then the last return is being processed, thereby allowing the entry in the stack to be removed.

39. Referring to claim 24, Gochee has taught a method as described in claim 18. Gochee has not taught that said detecting comprises a branch target buffer detecting the

call instruction in response to the call instruction being fetched from an instruction cache. However, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction. Furthermore, Official Notice is taken that instruction caches and branch target buffers (BTB) are well known and expected in the art. More specifically, a branch target buffer is indexed by the fetch address used to fetch the current instruction from the instruction cache. If the instruction is a branch, then it will have a corresponding entry in the BTB, thereby causing a hit to occur. In response to a hit, the BTB will provide a prediction (usually based at least on some sort of history). This BTB allows predictions to be made so that a system may enjoy the benefits taught by Hoyt. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include a BTB for detecting a call instruction and providing a prediction.

40. Referring to claim 25, Gochee has taught a method as described in claim 18. Gochee has not explicitly taught that said detecting comprises a decode unit detecting the call instruction in response to decoding the call instruction. However, it is inherent that a decoder exists within the system so that instructions may be decoded and the necessary control signals be sent to the execution units to perform the desired operations. Although there are multiple ways to detect a call instruction, Official Notice is taken that detection through decoding is well known in the art. By using the decoder to detect, additional hardware, such as a BTB, is not required to detect, thereby reducing required chip area.

As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee such that the decoder detects the call instruction.

41. Referring to claim 29, Gochee has taught a method as described in claim 27.

Further, claim 29 is rejected for the same reasons set forth in the rejection of claim 7.

42. Referring to claim 30, Gochee has taught a method as described in claim 29.

Further, claim 30 is rejected for the same reasons set forth in the rejection of claim 8.

43. Referring to claim 31, Gochee has taught a method as described in claim 29.

Further, claim 31 is rejected for the same reasons set forth in the rejection of claim 7.

44. Referring to claim 35, Gochee has taught a method comprising:

a) detecting a return instruction in an instruction stream and in response to said detecting, providing a first return address stored in a return storage and decreasing a count associated with the first return address. See Fig.7, steps 706-710.

b) Gochee has not taught that the provided return address is a predicted return address.

However, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction.

45. Claims 13 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gochee, as applied above, in view of "Stacks," downloaded from [www.funducom.com](http://www.funducom.com), April 2001 (herein referred to as Webinfo).

46. Referring to claim 13, Gochee has taught a mechanism as described in claim 1. Gochee has not explicitly taught that the first entry comprises a pointer to a next entry in the return storage. However, Webinfo has taught that a stack may be implemented as a linked list, which is more sophisticated than other implementations, and it allows for the pushing of as many elements as the user wants. See the “Implementing Stack as a Linked List” section. As can be seen from the code snippet in the same section, each node contains a link pointer which points to the next location in the stack. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee in view of Webinfo such that the stack is implemented as a linked list.

47. Referring to claim 34, Gochee has taught a method as described in claim 18. Further, claim 34 is rejected for the same reasons set forth in the rejection of claim 13.

48. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gochee in view of Hoyt, as applied above, in view of Tran, U.S. Patent No. 5,822,575 (herein referred to as Tran).

49. Referring to claim 15, Gochee has taught a mechanism as described in claim 1. Gochee has not taught a stack structure configured to store tags identifying entries in the return storage, wherein a top tag in the stack structure identifies the first entry. However, Hoyt has taught that predicting branches is desirable because it allows the system to continue fetching and providing instructions for execution as opposed to stalling, thereby improving efficiency. See column 1, lines 40-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee to include branch prediction. And, with a system using branch prediction, Tran has taught that

branch tags within each entry of a stack may be employed. See column 20, lines 54, to column 21, line 12. In essence, this is useful because when a branch misprediction occurs, the tag of the branch is broadcasted and entries within the return storage that are associated with call instructions with tags subsequent to the mispredicted call instruction's tag are discarded. This ensures that only instructions which should be executed are actually executed. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Gochee such that each stack entry has a tag, wherein a top tag in the stack structure identifies the first entry.

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH

David J. Huisman

September 1, 2004

*Eddie*

EDDIE CHAN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100